

Security Benchmarking Tool Documentation

Table of Contents

Security Benchmarking Tool Documentation	1
Table of Contents	1
Motivation	1
Preparing to Run the Application	1
Running the Application	2
Stopping the Application	2
Map	2
Excursion	4
Scenario	4
Civilian	5
Intruder	5
Waypoint	6
Behavior	8
Edge	9
Security Solution	12
Sensor	13
Security Solution Variation	17
Replicate	19

Motivation

The Security Benchmarking Tool (SBT) is a tool that allows security specialists and technologists (including sensor designers, developers of security operating procedures, behavior recognition algorithm researchers, etc.) to gain insight into the applicability of proposed solutions to base security problems. The tool focuses on behavior and the ability to detect and recognize it. It allows users to define notional scenarios where actors (simulation agents) emit specific behaviors, and then notional solutions which are intended to identify those behaviors which are “abnormal” in the context of the scenario. The solution can then be run against the scenario using the agent-based simulation engine which underlies the SBT. Analyzing the results of these runs can aid the user in evaluating the suitability of the proposed solution for the scenario.

Preparing to Run the Application

First, install the MATLAB Component Runtime on your computer by running mcrInstaller.exe. This is a library that enables the application to plot the results of your analysis.

Second, start the database backend by running startdb.bat. This allows the application to store Scenarios, Security Solutions, Excursions, and results for later use.

Running the Application

After you have installed the MATLAB Component Runtime and started the database, you are ready to run the application. Run the application by executing pfpl.exe.

Stopping the Application

You exit the application as you would any windows application, however you may want to stop the database after you are finished. This will not result in any work you have saved being lost. To stop the database, run stopdb.bat.

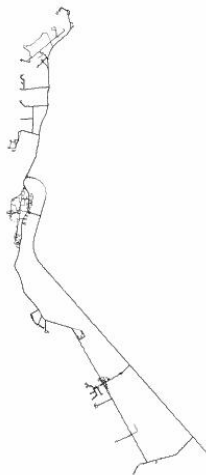
Map

The map is a vector based representation of the base that allows you to view and interact with agent [waypoint graphs](#), [behaviors](#), agents in a running [Replicate](#), and [sensors](#).

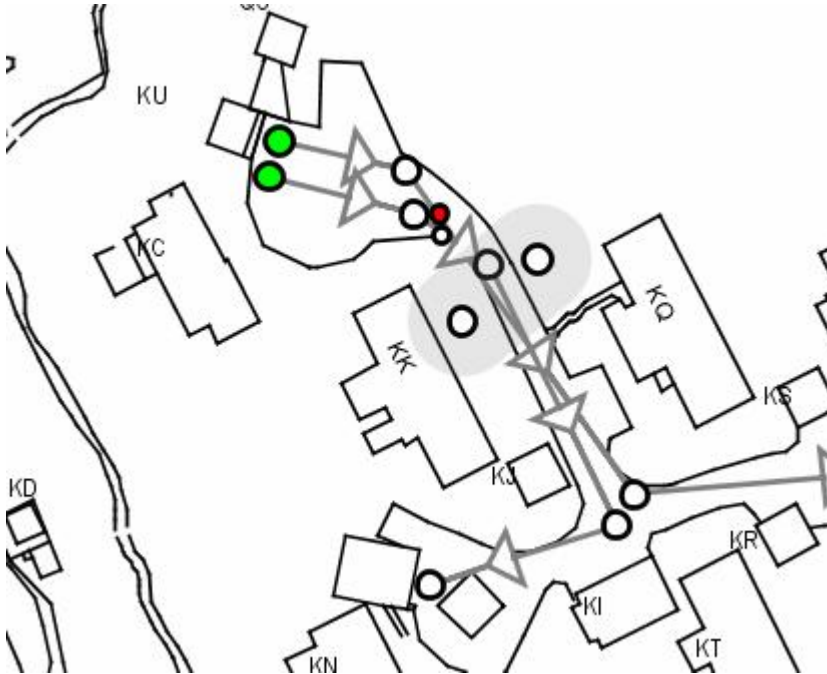
Zoom

You can zoom in and out of the map by holding down the Alt key on your keyboard while left click dragging your mouse to the right (to zoom in) or to the left (to zoom out). You will notice that the detail displayed by the graph changes for performance reasons depending how the map is zoomed. The farther you are zoomed in, the more detail you are able to see. Also notice that there is a limit to how far you can zoom in and out. The outward limit is set so that the map doesn't get so small that you are unable to find it again, and the inward limit is set at a reasonable viewing scale for the finest level of detail that the background can display.

Zoomed out:



Zoomed In (viewing a running [Replicate](#) with a [Civilian](#), an [Intruder](#), and a [Sensor](#)):



Pan

You can pan the map by holding down the Shift key on your keyboard while left click dragging your mouse in any direction.

Panning to an Agent or [Sensor](#)

When you click on an agent or sensor in the simulation tree, the map will automatically make the agent or sensor visible if it is not already and pan to center itself on the closest waypoint or control point for that agent or sensor, respectively. This enables quick and convenient navigation from items in the simulation tree with items as they are displayed in the map.

Viewing a Running Replicate

As a [replicate](#) is running, you can observe agents moving from point to point in the map. [Intruders](#) are depicted in red, while [Civilians](#) are depicted in white. It takes a significant amount of processing capacity to update the display as the agents move from point to point, so you may see the simulation slow a bit as you are watching it. If you are satisfied with how the agents are moving through the map and would rather not watch the simulation, simply pan away so that there are no agents displayed, and the simulation will then run at full speed since it will not need to do any graphics updates.

Excursion

A Scenario is a set of [Civilian](#) and [Intruder](#) agents that have scripted movement and behaviors over time within the context of a simulation. As a simulation is run (a run of a simulation is called a [Replicate](#)), the [Security Solution](#) attempts to categorize each agent as either a [Civilian](#) or an [Intruder](#) based on behaviors that are observed its [Sensors](#).

Loading a Scenario

Loading a Security Solution

Creating a new Security Solution Variation

Properties

Name	The name of the Excursion.
------	----------------------------

Operations

Load Scenario	Replace the current Scenario with one that is stored in the database. You will be presented you with a dialog where you can choose the new Scenario from the database by name or cancel the operation.
Load Security Solution	Replace the current Security Solution with one that is stored in the database. You will be presented you with a dialog where you can choose the new Security Solution from the database by name or cancel the operation.
Save	Save this Excursion. This will save the Scenario , Security Solution , Security Solution Variations , and Replicates of this Excursion as well.
Copy As...	Create a copy of the Excursion with a new name. Note that this will make an exact copy of everything under the Excursion in the simulation tree. This includes currently loaded Scenario and the currently loaded Security Solution (keeping the same name for these objects), Security Solution Variations , and Replicates and results from running Replicates .

Scenario

A Scenario is a set of [Civilian](#) and [Intruder](#) agents that have scripted movement and behaviors over time within the context of a simulation. As a simulation is run (a run of a simulation is called a [Replicate](#)), the [Security Solution](#) attempts to categorize each agent as either a [Civilian](#) or an [Intruder](#) based on behaviors that are observed its [Sensors](#).

Loading a Scenario

See the Loading a Scenario section in [Excursion](#).

Adding Civilian Agents

Adding Intruder Agents

Properties

Name	The name of the Scenario.
Stop Time	The time at which the Scenario should stop running. This is set to 1,000 timesteps by default. This determines how long any Replicates that run against this Scenario will run.

Operations

Save	Save the Scenario to the database. The scenario can then be retrieved later from the same database using its name.
Copy As...	Save the Scenario to the database. The scenario can then be retrieved later from the same database using its name.
Plot Scenario	Produces a histogram displaying, for each Security Solution associated with the scenario, the number of Intruders identified, the number of Civilians as Intruders, the number of Civilians identified, and the number of Intruders as Civilians. When Plot Scenario is selected an input box appears requesting input of the threshold to use in the evaluation.

Civilian

Civilians are simulation agents that represent people who are not [Intruders](#). Civilians include residents of the local area, base officials, base employees, and military personnel. Civilians can move and emit behaviors in the simulation. A Civilian moves along a series of [Waypoints](#) on the map. At each [Waypoint](#) the user can select a series of destination [Waypoints](#) and assign a [weight](#) to each which determines the likelihood that that [Waypoint](#) will be the agent's actual destination. The user can also assign a [speed](#) at which the agent will travel to the next [Waypoint](#).

Intruder

Intruders are simulation agents that represent people who have malicious intent toward the base. Intruders can move and emit behaviors in the simulation. An Intruder moves along a series of Waypoints on the map. At each [Waypoint](#) the user

can select a series of destination [Waypoints](#) and assign a [weight](#) to each which determines the likelihood that that [Waypoint](#) will be the agent's actual destination. The user can also assign a [speed](#) at which the agent will travel to the next [Waypoint](#).

Waypoint

A waypoint indicates a point at which an agent will change direction or behavior. Agents travel from waypoint to waypoint along [edges](#) that connect waypoints, and perform behaviors as they travel as directed by their waypoints.

Creating new waypoints

You can create a new waypoint by right-click dragging an existing waypoint to an empty space on the map.

Each agent has its own set of waypoints and starts at the green waypoint, called the starting point. A detailed example of this is given in the [Drawing an Edge](#) section of the [edge](#) documentation.

Moving Waypoints

You can change the location of a waypoint by left-click dragging it on the map.

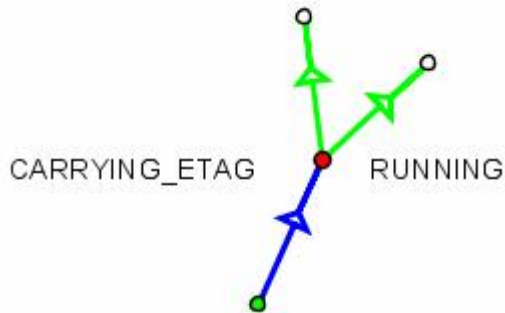
Selecting Waypoints

You select waypoints by left-clicking on them. After you select a waypoint, the waypoint will turn red, and any incoming edges will turn blue, and incoming edges will turn green. You will also be able to see any behaviors associated with the waypoint. For example, the center node in the following picture is selected and specifies a single "RUNNING" behavior:



Adding Behaviors

You can add a new [behavior](#) after you have selected a waypoint by right-clicking on the waypoint and selecting “Add Observable Behavior”. You can add multiple behaviors to the same waypoint by repeating the process. For example, now the center node specifies both “RUNNING” and “CARRYING_ETAG”:



You can change properties of the behavior to be emitted after you create it by left clicking on the text of the behavior:



The behavior will turn red, and you will see the properties of the behavior appear in the property inspector:

Properties	
Characteristics	
Type	CARRYING_ETAG
Emit Each Timestep	<input type="checkbox"/>
Probability	1

The [documentation for behavior](#) has detailed information on the meaning of these properties.

Duplicating Behaviors

If you right click drag a waypoint to create the next waypoint, the new node will be given the same set of behaviors as the original. This becomes convenient when creating many waypoints where the behavior does not change.

Deleting Waypoint

To delete a waypoint, first select it and then press the delete key on the keyboard. The waypoint will be deleted, as well as any incoming or outgoing edges adjacent to the waypoint that is deleted.

Hiding a Waypoint Graph

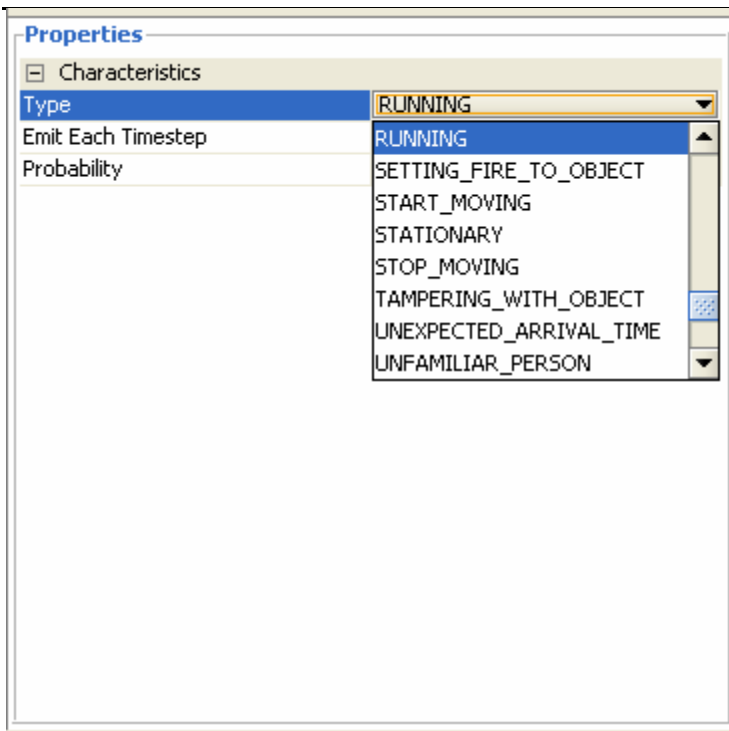
When there are many agents and sensors, sometimes the screen may begin to become cluttered. You can hide a waypoint graph by right clicking on a waypoint, edge, or behavior in the graph and then picking “Hide”. This will remove the graph from the view, and the graph can be restored to the view by clicking on the corresponding agent in the simulation tree.

Behavior

Observable behaviors are actions or expressions of state emitted by [Civilian](#) and [Intruder](#) agents that can potentially be recognized by [Sensors](#). Each agent can emit one or more behaviors, either once when it reaches a [Waypoint](#), or at every timestep along an [Edge](#) between [Waypoints](#). Examples of behaviors are “behaving nervously”, “running”, “familiar person”, and “carrying explosives”.

To have the agent emit the behavior at each timestep between the current and the next waypoint, check the Emit Each Timestep property. If the property is unchecked, the behavior will be emitted only at the timestep the agent reaches the waypoint.

The Probability attribute indicates the probability that the agent will emit the behavior. In the case where Emit Each Timestep is checked, whether the agent emits the behavior or not is determined independently at each timestep between the current and the next waypoint.



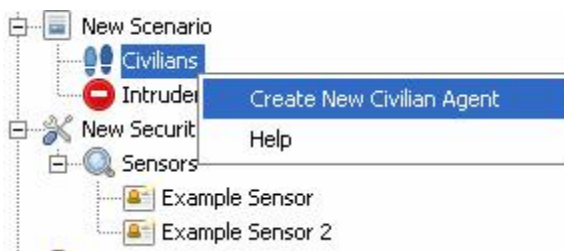
Edge

The edge of the agent movement determines the path upon which an agent will move from [waypoint](#) to [waypoint](#) during the run of a [Replicate](#).

Drawing an Edge

The following example will demonstrate creating new edges.

1. Right click on the [Civilians](#) item in the simulation tree and select “Create new Civilian Agent”



Enter a name for the agent in the dialog that appears and then press OK.

2. A green circle appears in the map view. This is the starting point for the agent's waypoint graph. Right click on the starting point and drag to another point on the map then release the right mouse button. Your agent's waypoint graph should now look something like this:



The arrow between the two [waypoints](#) is the edge, and if the simulation was run at this point, the agent would move from the green [waypoint](#) (the starting point) to the white one along the direction of the arrow.

Selecting an Edge

To select an edge, select the agent, either by clicking on it in the simulation tree or by left clicking on one of the nodes. Then click on the edge you would like to select. The edge should turn red, like this:



Deleting Edges

To delete an edge, select it and then press the delete button. The edge will be deleted, but the nodes will remain:

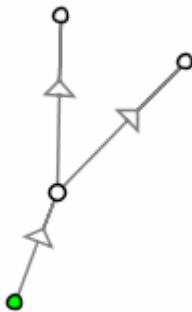


We can reconnect the two nodes with a new edge by right click dragging from one to another:



Assigning Probabilities to Edges

The probability of an edge determines the likelihood that the agent will follow the edge if it has an option between two edges. For example, in a waypoint graph like this one:



When the agent gets to the center [waypoint](#), it has the option of taking the left edge, or the right one. You can assign probabilities to each edge by clicking on each edge in turn and changing the “probability” property in the property editor. When the agent determines which path to take, the absolute value of the probabilities is not important, it is the ratio of the probabilities. For example, if the probability property is set to 1 for the left path and 2 for the right, then the agent will be twice as likely to take the right as the left. The same is true if the left value is 5 and the right is 10.

Assigning Speed to Edges

The relative speed at which the agent travels along an edge can be changed by setting the “speed” property of an edge. The default value for “speed” is 0.1.

Assigning Path Variance to Edges

Assigning a non-zero path variance will cause the agent to wobble randomly along the path of an edge. Larger values indicate a wobble of greater magnitude.

Hiding a Waypoint Graph

When there are many agents and sensors, sometimes the screen may begin to become cluttered. You can hide a waypoint graph by right clicking on a node or edge in the graph and then picking “Hide”. This will remove the graph from the view, and the graph can be restored to the view by clicking on the corresponding agent in the simulation tree.

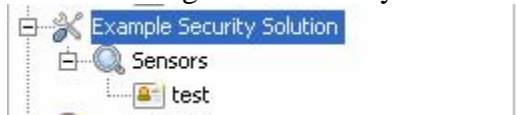
Security Solution

A security solution is a set of [Sensors](#) that detect the [behaviors](#) of [Civilian](#) and [Intruder](#) agents of a [Scenario](#) during runs of an [Excursion](#) (called [Replicates](#)). The [behaviors](#) that are detected by the security solution are used by the system to assign a [threat level](#) for each agent detected at each timestep that the agent is detected. This [threat level](#) is stored in [Observed Agent](#) items as the [Replicate](#) runs. The performance of a security solution can be judged against various [Scenarios](#) by evaluating how accurately and how early the system is able to differentiate [Civilians](#) from [Intruders](#) based on assigning a higher threat level to [Intruders](#) than [Civilians](#).

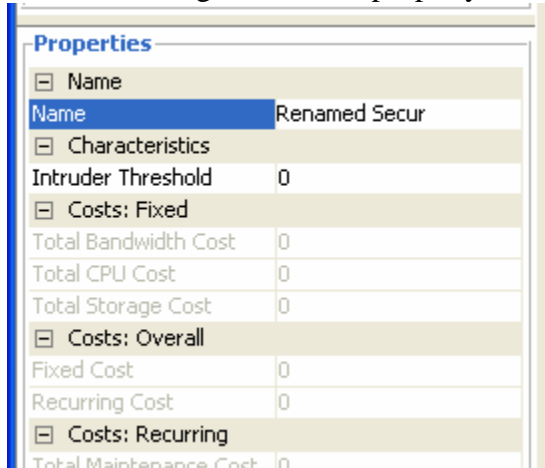
A security solution also has costs associated in it. These costs are summations of the costs of the security solution’s component [Sensors](#). Generally speaking, it is desirable for a security solution to minimize costs while continuing to perform well against a wide range of [Scenarios](#).

Changing the name of a Security Solution

You can change the name of your Security Solution by selecting it in the simulation tree:



and then editing its “Name” property in the property editor:



Saving a Security Solution

You can save a Security Solution by right-clicking on the Security Solution and choosing “Save”. Your Security Solution will be saved to the database from which it can be retrieved later by its name using the [Load Security Solution](#) operation of the [Excursion](#).



Copying a Security Solution

By right-clicking on the Security Solution and choosing “Copy As...”, you can create a copy of your Security Solution in the database with a new name. All of the Sensors of your Security Solution will also be copied. When you choose “Copy As...” you will be prompted by a dialog to provide a name for the copy:

Adding New Sensors

See the [Creating a Sensor](#) portion of the [Sensor](#) documentation.

Plotting

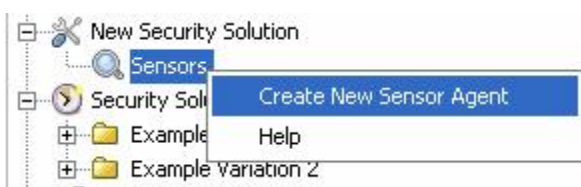
By right-clicking on the Security Solution and choosing “Plot Solution”, you can display a histogram displaying, for each Scenario associated with the Security Solution, the number of Intruders identified, the number of Civilians as Intruders, the number of Civilians identified, and the number of Intruders as Civilians. When “Plot Solution” is selected an input box appears requesting input of the threshold to use in the evaluation.

Sensor

Each Sensor defines a [coverage area](#) on the map where there is a [probability of detecting](#) agents performing a [certain behavior](#). Each sensor also has associated [fixed](#) and [recurring](#) costs and a [false alarm rate](#).

Creating a Sensor

To create a new Sensor, right click on the Sensors item in the [Security Solution](#) and select “Create New Sensor Agent”:

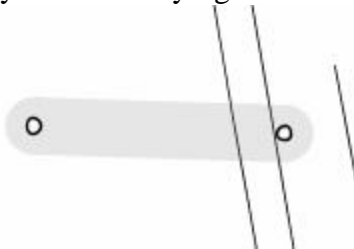


Your new Sensor will be created and then selected. You can see the location of your new Sensor as a white circle in the middle of the map:



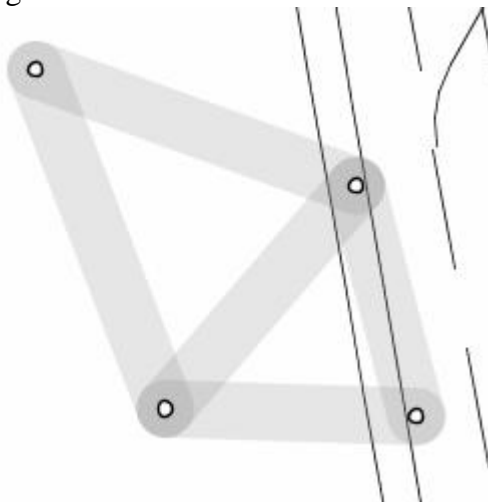
Specifying the Sensor's Coverage Area

This white circle is the first control point for your sensor. You can define a coverage area for your sensor by right-click dragging this control point to another location on the map:



The transparent gray area is the coverage area, the area where behaviors may be detected.

You can create an unlimited number of control points for your sensor to define irregular coverage areas:

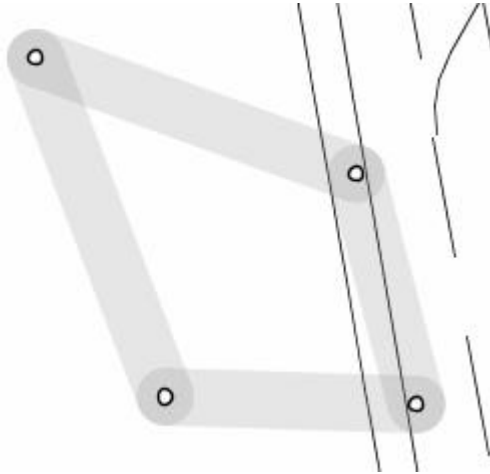


Note that in the above example, even though some areas of the coverage area appear darker above, there is an equal [probability of detection](#) for every point in the coverage area.

Also note that multiple sensors are allowed to have overlapping coverage areas.

Deleting Coverage Area

To reduce the amount of coverage area of your Sensor, you can delete either control points, or the actual areas connecting the control points. To do this, left-click on the item you would like to delete, and then press the delete key on the keyboard. To illustrate, in the example above, if you click in the gray diagonal portion of the coverage area, and then press the delete key, we can remove a portion of our sensor. Our sensor now looks like this:



Sensor Properties

After a sensor is selected, you can view and edit its properties in the property editor:

Properties	
[-] Costs: Fixed	
Bandwidth Cost	0
CPU Cost	0
Storage Cost	0
Total Fixed Cost	0
Fixed Unit Cost	0
[-] Characteristics	
Behavior Detected	RUNNING
Coverage Width	2
False Alarm Rate	0.0001
Probability of Detection	1
[-] Costs: Recurring	
Maintenance Cost	0
Operational Cost	0
Total Recurring Cost	0
Training Cost	0
Recurring Unit Cost	0
[-] Name	
Name	Example Sensor

Behavior Detected

This is the [behavior](#) that the sensor has the capability to detect.

Probability of Detection

The probability of detection is a value between 0 and 1 inclusive that indicates the probability that the “[Behavior](#) Detected” [behavior](#) will be sensed by the sensor when the [behavior](#) is performed within the sensor’s [coverage area](#). This probability is evaluated once per agent per timestep that the behavior is emitted within the [coverage area](#).

False Alarm Rate

The false alarm rate is a value between 0 and 1 inclusive that indicates the probability that an agent in the sensor’s [coverage area](#) will be sensed as performing the “[Behavior](#) Detected” behavior whether or not the agent in the [coverage area](#) is actually performing the behavior. This probability is also evaluated once per agent per timestep when agents are in the [coverage area](#).

Name

The name by which the sensor is identified.


Fixed Costs

Bandwidth Cost, Storage Cost, and CPU Cost are each multiplied by the Fixed Unit Cost, and then summed in order to calculate the Total Fixed Cost.

Recurring Costs

Maintenance Cost, Training Cost, and Operational Cost are each multiplied by the Recurring Unit Cost, and then summed in order to calculate the Total Recurring Cost.

Total Costs

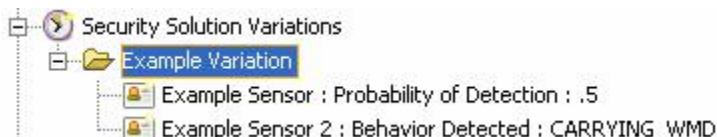
The total costs are aggregates of the different cost categories, and are used in calculating the overall cost of the Security Solution, which can be viewed in the property editor by clicking on the Security Solution  **New Security Solution** item in the simulation tree. The costs are computed and stored for each [Replicate](#) so that cost trades can be evaluated against [Security Solution](#) Performance by applying various [Security Solution Variations](#).

Security Solution Variation

A Security Solution Variation is a list of modifications Sensors in a Security Solution. The modifications are specified as a list of triplets, each made up of a Sensor name, a name of a property to change, and the new value of that property.

A Security Solution Variation gives you a way to modify properties of interest in a Security Solution across different Replicates of that Security Solution. For example, you might be interested in how changing the Probability of Detection property of a particular sensor affects the efficacy of a Security Solution. When a Replicate that corresponds to the Security Solution is run, the Security Solution is applied first, so that the effects of the varied Sensor properties can be observed.

For example, the following Security Solution Variation makes a modification to a property of “Example Sensor” and to a property of “Example Sensor 2”:



The following graphic shows an example of one triplet in the example Security Solution Variation:

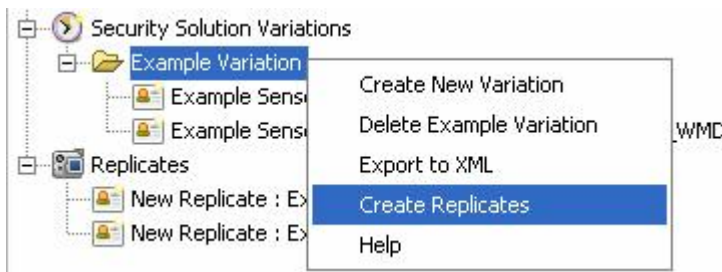
Properties	
Name	
Sensor Name	Example Sensor
Property Name	
Property Name	Probability of Detection
Value	.5

In this case, the value of the “Probability of Detection” property of the sensor named “Example Sensor” will be changed to .5 from whatever its value is specified in the base Security Solution. This change will be made immediately before any replicate of this

Security Solution Variation is run, and reversed to its base value when the Replicate finishes or is stopped.

Create Replicates

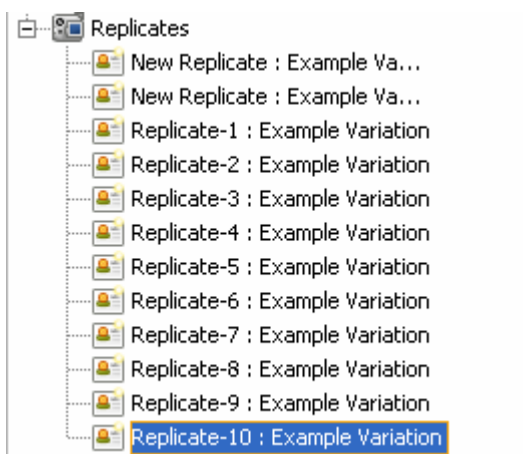
You can create a number of Replicates for one Security Solution Variation by choosing the “Create Replicates” option for that Security Solution Variation:



Once making this selection, you will be asked how many Replicates you would like to create:



and then the appropriate number of Replicates will be added to the end of the Replicates List:

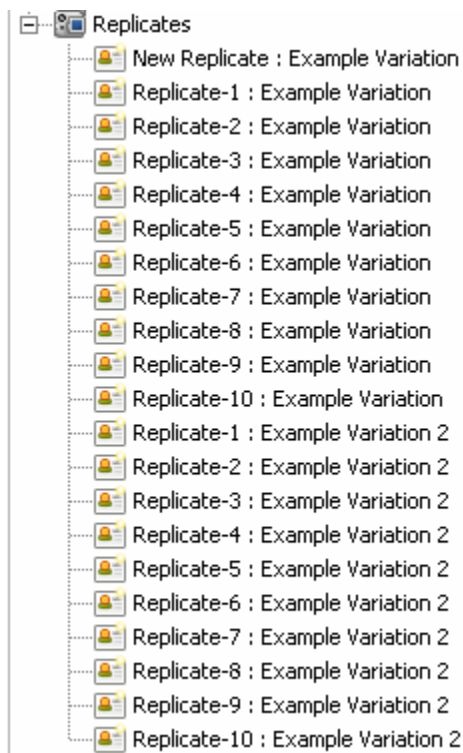


Create Replicates For All Variations

If you have a large number of Security Solution Variations to investigate, you can create a specified number of Replicates for all of them by selecting the “Create Replicates For All Variations” option from the  Security Solution Variations item:



You will again be prompted for the number of replicates to create, and the corresponding Replicates will be added to the simulation tree:




You can run all the replicates you have just created by following the documentation in the “Run All Replicates” section of the documentation for [Replicate](#).

Replicate

A replicate is a run of an [Excursion](#) with a particular and random number seed. The output of a Replicate that has been run is a list of [Agent Observations](#) that are

occurrences of [Scenario](#) Agents being detected by the [Security Solution](#) and assigned a threat level.


Creating a Replicate

To create a new Replicate, right-click on the Replicates node  Replicates in the simulation tree and select “Create New Replicate”. The new Replicate that was created will then be selected in the tree.

Deleting a Replicate

Deleting a Replicate removes all information associated with a replicate from the database, including information about past runs of the Replicate.

Running a Replicate

To run a Replicate, select it in the Simulation Tree, and then click the play button  that appears at the bottom of the window below the Map display.

Before a Replicate runs, the Replicate’s [Security Solution Variation](#) is applied to the current [Security Solution](#). After the Replicate finishes, the [Security Solution](#) is returned to its previous state.


As a replicate runs, it counts through the timesteps of the [Scenario](#), starting from 0 and ending at the [Stop Time](#) of the [Scenario](#). At each timestep, 3 things occur in order:



1. The [Civilians](#) and [Intruders](#) move and emit behaviors for that timestep.
2. The Sensors try to observe behaviors of any Civilians or Intruders in their coverage area.
3. The system assigns a threat level to each Agent that was observed by one or more Sensors based on the Behaviors that were observed at that timestep.

The output of this process is summarized in [Observed Agent](#) items that are populated within the Replicate as it is running. The movement of [Civilians](#) and [Intruders](#) is displayed on the map as the Replicate runs.


Replicates are repeatable, that is, assuming the [Scenario](#), [Security Solution](#), [Security Solution Variation](#), and random number seed have not changed, re-running the Replicate will always produce the same results.

Pausing a Replicate


To pause the currently running Replicate, click on the Replicate (or any other Replicate) and then click the pause button . When the Replicate is paused, you can step the

simulation by one timestep at a time using the step button . You un-pause the simulation (continue the run) by pressing the pause button  again.

Stopping a Replicate

To stop the currently running Replicate before the [Stop Time](#), click on the Replicate (or any other Replicate) and then click the stop button .

Repeating a Replicate

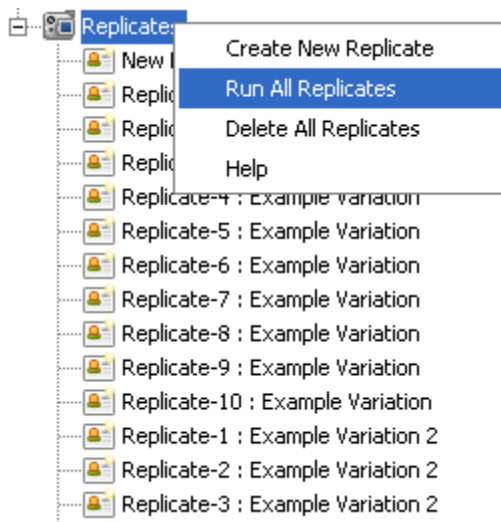
To repeat a Replicate, first stop any running Replicates, and then click on the replicate you would like to re-run. Then press the play button . Any [Observed Agents](#) that were recorded during the previous run will be deleted as the Replicate starts.

Creating Replicates for Batch Runs

See the instructions for [Create Replicates For All Variations](#) and/or [Create Replicates](#) in the [Security Solution Variation](#) documentation.

Run All Replicates

You can perform a batch run of all Replicates in an [Excursion](#) by right-clicking on the  Replicates node and choosing “Run All Replicates”:



You will then see a progress bar appear that shows how the batch run is progressing and allows you to cancel the run:



When running Replicates in batch, the shorter the [Stop Time](#) of your [Scenario](#), the faster your Replicates will run.

Note that by choosing Run All Replicates, you will lose any previously computed results for the list of Replicates.

Delete All Replicates

This operation deletes all replicates and their results from your [Excursion](#).

Properties

Name	The name of the Civilian or Intruder agent that was observed.
Random Number Seed	The random number seed assigned to this Replicate.
Security Solution Variation	The timestep at which the observation was made.
Total Bandwidth Cost	The sum of the Bandwidth Cost for the Security Solution.
Total CPU Cost	The sum of the CPU Cost for the Security Solution.
Total Storage Cost	The sum of the Storage Cost for the Security Solution.
Total Maintenance Cost	The sum of the Maintenance Cost for the Security Solution.
Total Operational Cost	The sum of the Operational Cost for the Security Solution.
Total Training Cost	The sum of the Training Cost for the Security Solution.
Fixed Cost	Total Bandwidth Cost + Total CPU Cost + Total Storage Cost
Recurring Cost	Total Maintenance Cost + Total Operational Cost + Total Training Cost

This product includes JCommon (<http://www.jfree.org/jcommon/index.php>).

This product includes Jess (<http://www.jessrules.com/>).

This product includes Hibernate (<http://www.hibernate.org/>).

This product includes JUnit (<http://www.cs.wm.edu/~noonan/junit/>).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes MASON (<http://cs.gmu.edu/~eclab/projects/mason/>).

This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>).